84K00XXX
Rev. BASIC

Checkout and Launch Control System (CLCS) Juno Delivery

System Test Procedures Document

April 14, 1997

Prepared By:

CLCS System Test Team
CLCS Technical Integration
DE-CLC
Kennedy Space Center, FL 32899

Checkout and Launch Control System (CLCS) Juno Delivery

System Test Procedures Document

Prepared By:

David Reinhardt, System Test Engineer	Timothy McKelvey, System Test Engineer

Approved By:

Arnold Postell, Technical Integration Lead	Kirk Lougheed, System Engineering and Technical Integration Lead

Table of Contents

1.0 Scope

This document defines the test approach and procedures to be executed for the Juno delivery by CLCS System Engineering and Technical Integration. For the Juno delivery, testing will occur at Kennedy Space Center in the Engineering Development Laboratory (EDL) Satellite Development Environment (SDE) as well as in the Launch Control Center LCC-X demonstration facility.

1.1 Identification

This document is the Checkout and Launch Control System (CLCS) Juno Delivery System Test Procedures Document, Rev. BASIC.

1.2 Purpose

The purpose of this document is to define a suite of test procedures that will accurately assess the delivered hardware and software to ensure it is functional and meets project commitments for the Juno delivery. This Juno delivery System Test will not result in formal “buy-off” of any requirements, therefore Quality Assurance sign-off is not required. This test will act as a path finder to ensure that the CLCS delivery process is understood and complete. Formal system testing will begin with the Redstone Delivery currently scheduled for September, 1997.

1.3 System Overview

The CLCS is composed of a Real Time Processing System (RTPS) and a support system called the Shuttle Data Center (SDC). The RTPS provides the capability to checkout and control the elements of the current Space Transportation System (STS) and Ground Support Equipment (GSE). It provides support to the Space Shuttle Program into the 21st Century and a basic infrastructure upon which to base future design projects such as the Orbiter Upgrades and RLV.

The CLCS replaces the current Launch Processing System (LPS) with state-of-the-art Commercial Off the Shelf (COTS) based technology. Wherever possible, COTS software is used instead of developing software where the requirements of the task are satisfied by the COTS software products. Any developed software will be written in high level languages that have demonstrated a high degree of portability between platforms. COTS hardware is also utilized as much as possible in the CLCS. This strategy provides a highly reliable system that is both supportable, with minimal software and hardware upgrade impacts, and expandable, with a solid base design.

1.4 Hardware and Software Configurations

Figure 1 shows the Juno hardware configuration for the LCC-X demonstration facility. Figure 2 shows the Juno hardware configuration for the Satellite Development Environment 1 (SDE1) in the EDL. This test will utilize a portion of each of these environments. Specific equipment to be used will be documented in the test procedures at test time.

Figure 3 shows the Juno software configuration in the SDE1. Not all of the software in SDE1 is included in the Juno baseline. See Table 1 for a listing of the baselined software. Figure 4 shows the software configuration in the LCC-X demonstration facility. The LCC-X demonstration facility is not intended to be part of the baselined CLCS, therefore the configuration will vary depending on planned demonstrations (e.g., new versions of CLCS S/W, new HCI technology, new COTS capabilities). For the Juno System Test, the LCC-X software will be configured similarly to the SDE1 (given the differences in H/W availability)

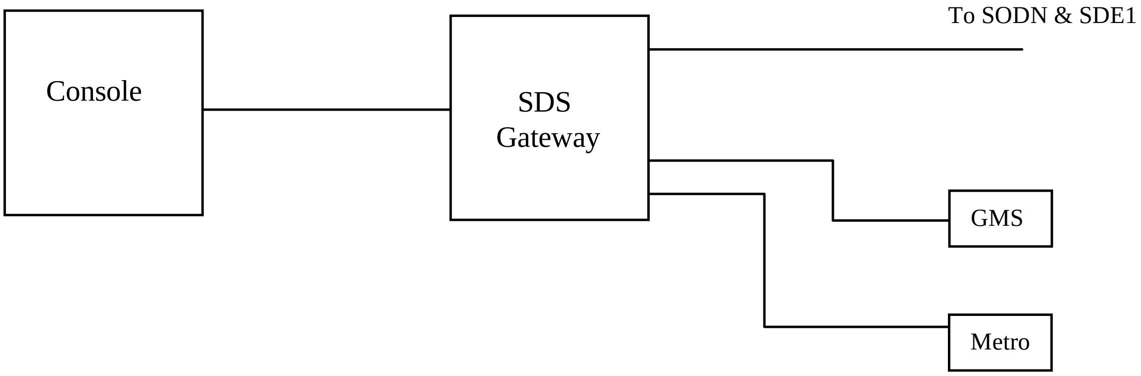
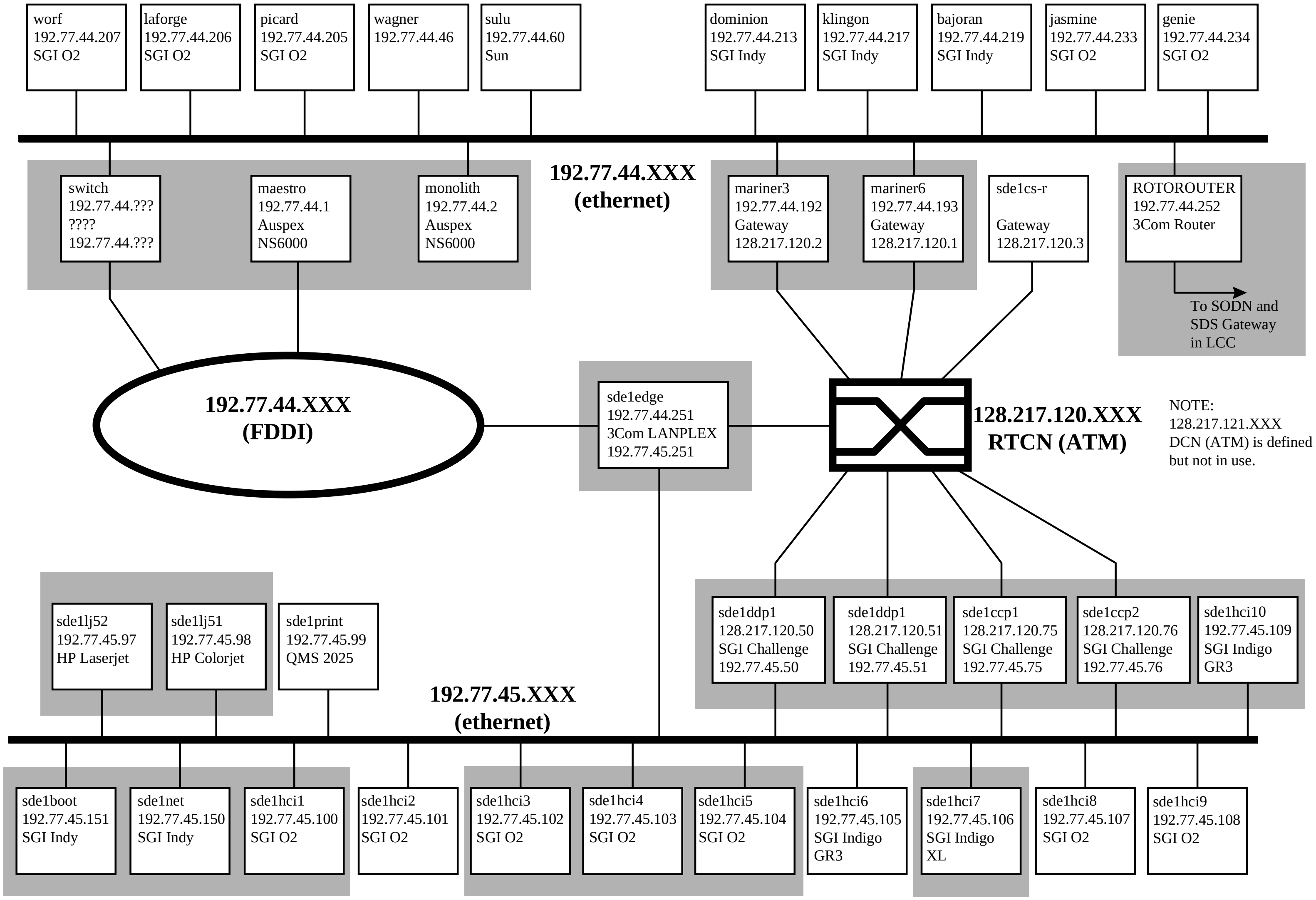


Figure 1: LCC-X Hardware Configuration
DRAWING NOT CORRECT YET



NOTE:
128.217.121.XXX
DCN (ATM) is defined
but not in use.

Figure 2: Satelite Development Environment - SDE1
(Shaded areas indicate equipment required for System Test)

Figure 3: SDE1 Software configuration <TBS>

Figure 4: LCC-X Software configuration <TBS>

Software	Vendor	Version	Platform	Facility
IRIX (UNIX operating system)	Silicon Graphics Incorporated (SGI)	6.2	SGI Indigo 2, SGI Indy, SGI Challenge	SDE, LCC-X
IRIX (UNIX operating system)	Silicon Graphics Incorporated (SGI)	6.3	SGI O2	SDE, LCC-X
VxWorks (Gateway OS)	VxWorks	5.2	SDS Gateway, CS Gateways	SDE, LCC-X

Table 1.1: Juno Baselined COTS Software

Software Module	CSCI	Version	Platform	Facility
			SGI Indigo 2, SGI Indy, SGI Challenge	SDE, LCC-X
			SGI O2	SDE, LCC-X
			SDS Gateway, CS Gateways	SDE, LCC-X

Table 1.2: Juno Baselined Custom Software

1.5 Document Organization

This document is divided into three sections and four appendices:

Section 1, Scope, discusses the purpose of the CLCS System Test, provides a system overview , and describes software and hardware configurations for the system.

Section 2, Applicable Documents, lists the documents used to create and those supporting this document.

Section 3, Test Case Description, contains a description of the test cases, the pass/fail criteria, and the procedures in detail.

Appendix A, Acronyms and Definitions, contains a listing of acronyms and selected word definitions (for words which may have multiple interpretations)

Appendix B, Requirements Traceability and Test Methods Matrix, contains the requirements verification matrix for the test.

Appendix C, Resource Requirements, contains a list of software, hardware, and personnel requirements necessary for each test.

Appendix D, Standard Operating Test Procedures, contains any specific, standard procedures identified within the test cases.

2.0 Applicable Documentation

The following documents, of the revision shown, form a part of this document to the extent specified.

2.1 Parent Documents

The documents in this paragraph establish the criteria and technical basis for the existence of this document. The parent documents are:

Parent Document	Document Number	Rev.	Date
CLCS System Test Plan	N/A[1]	BASIC	3/27/97
CLCS Program Management Plan	*N/P*[2]	-	-
CLCS System Engineering Management Plan (SEMP)	*N/P*	-	-
CLCS Project Plan	*N/P*	-	-

Table 2.1: Parent Documents

2.2 Applicable Documents

Applicable documents are those documents which form a part of this document. These documents, at the revisions listed below, carry the same weight as if they were stated within the body of this document.

Applicable Document	Document Number	Rev.	Date
CLCS System Level Specification	*N/P*	-	-
CLCS CM Plan	*N/P*	-	-
CLCS Integration Management Plan	*N/P*	-	-

Table 2.2: Applicable Documents

[1] N/A indicates document number is not available. The configuration management process for CLCS program documents has not been finalized, however this document is considered baselined.

[2] *N/P* indicates that the document has not yet been published. References are included in these tables to show current expectations.

2.3 Reference Documents

Reference documents are those documents which, though not a part of this document, serve to clarify the intent and contents of this document.

Reference Document	Document Number	Rev.	Date
CLCS Certification Plan	*N/P*	-	-
CLCS System Design Document	*N/P*	-	-
CLCS Software Development Plan	*N/P*	-	-
CLCS Concept of Operation	*N/P*	-	-

Table 2.3: Reference Documents

3.0 Test Case Description

This section describes each test case, the expected results, the pass/fail criteria, and a step by step procedure to execute the test. Appendix B contains the Requirements Traceability and Test Methods Matrix, which maps system requirements to the test case that verifies those requirements. Test cases are stand alone, and can be executed in any order, however, it is recommended to run the test cases in the order in which they are presented.

3.1 System Integration Build and Configuration Management

The purpose of this test case is to demonstrate that the process for compiling, linking, and installing/distributing software is correct and consistent at the system level, as is the process for baseline control. These capabilities were developed in conjunction with the Juno Development Environment Foundation thread.

3.1.1 Test Description

3.1.1.1 Detailed description

Develop a short segment of code and use subsystem defined build procedures and any additional system level build procedures to take developed source code through to an operational configuration.

Verify that the software configuration control tool allows check-in and check-out of files and maintains a record of versions and changes.

3.1.1.2 Resource Requirements

3.1.1.2.1 Test Personnel

Personnel required include at a minimum a Test Conductor. Since this test will not result in formal “buy-off” of any requirements, a QA representative is not required. Skills required by test conductor (or designee) include knowledge of vendor operating procedures, including both standard user and system administrator procedures, and familiarity with software CM tools and procedures.

3.1.1.2.2 Hardware

The following hardware is required:

- Two Development workstations

3.1.1.2.3 Software

The following Software is required:

- Vendor supplied operating systems and associated utilities
- CLCS software configuration control tools (CVS)

3.1.1.2.4 Data

There are no specific data requirements for this test.

3.1.1.3 Requirements Trace

This test will simulate the demonstration of compliance with System Level Specification requirements covering the following functions:

- Create software
- Control software development
- Basic OS commands/services:
  - File copy
  - Text editing
  - Compile, make
  - Log-on, log-off

3.1.2 Pass/Fail Criteria

Successful completion of the test procedures without any problems that would result in the generation of critical PR's and without an excessive number of major PR's will be sufficient for this test to be considered passed.

3.1.3 Procedure

Refer to Procedure 3.1 in Table 3.1 for test procedures.

Procedure 3.1 - System Integration Build and Configuration Management

Date: **Location:** **Start Time:**

Test Setup/Initial Conditions - This test will be executed in the SDE1. All required software and hardware will be available in the same configuration they would be if the system were to be supporting daily software development. Two workstations will be powered up and ready to accept logins for a development session.

Step	Description	Expected Results	Comments	TC	QA
1.	Login as a software developer	Login is successful, initial user display is presented.			

2.	Start the text editor vi and enter the following code:
	 #include <stdio.h>
	void main(void)
	{
	 printf("Hello world\n");
	}
		Code is entered			

3.	Save code as file "test1.c"	File is saved			
4.	Compile code. Type: cc -o test1 test1.c	Compiles without error			
5.	Execute program test1. Type: test1	"Hello world" is displayed on screen			
6.	Using the xedit editor, write a make file for test1. Type: xedit (The file should contain the following lines) test1: ./test1.c <tab>cc ./test1.c -o ./test1 Save the file with the filename Makefile, quit xedit	The file "Makefile" is created.			
7.	Delete program test1. Type: rm test1	The file "test1" is deleted.			

Table 3.1

Step	Description	Expected Results	Comments	TC	QA
8.	Execute the Make file. Type: make	test1 compiles successfully			
9.	Execute program test1. Type: test1	"Hello world" is displayed on screen			
10.	Copy program test1 to another workstation: rcp test1 <workstation_id>:/tmp/test1	The program is copied to the remote workstation			
11.	logon to <workstation_id> and execute test1. Type: rlogin <workstation_id> /tmp/test1	"Hello world" is displayed on screen			
12.	Delete the program test1 and logout of the remote workstation. Type: rm /tmp/test1 exit	The file "/tmp/test1" is deleted and the remote login session is terminated.			
13.	Prepare the CVS/RCS repository by typing the following commands: cd ~ mkdir SW-REPOSITORY mkdir SW-REPOSITORY/Repository	The SW-REPOSITORY directory is created in the home directory and contains the Repository subdirectory.			
14.	Set up the required CVS environment variable, type (all on one line): setenv CVSROOT $HOME/SW-REPOSITORY/Repository	The environment variable is set.			

Table 3.1 (cont.)

Step	Description	Expected Results	Comments	TC	QA
15.	Initialize the repository by typing the following command: cvs init	CVS initializes without error message			
16.	Create a local working directory and copy the files created in steps 2 and 6 into it. Also copy two of the IRIX provided C files into the local working directory. Add the copied files to the repository. Type: mkdir $HOME/Working1 cd $HOME/Working1 cp /usr/include/strings.h . cp /usr/include/time.h . cp ../Makefile . cp ../test1.c . cvs import -m "Initial entry into CVS." JunoTest CLCS Start	The files will be added to the CVS repository under the JunoTest subdirectory			
17.	Checkout the file test1.c by typing the following command: cvs checkout JunoTest/test1.c	The file test1.c is available in the JunoTest subdirectory			
18.	Display the status the CVS repository by using the following command: cvs status	CVS reports the initial version (1.1) of the file test1.c			

Table 3.1 (cont.)

Step	Description	Expected Results	Comments	TC	QA
19.	Modify the test1.c file by adding a second printf line after the first: printf("Hello world again\n"); You may use vi or xedit to perform this task. The file is located in the JunoTest subdirectory (i.e. vi JunoTest/test1.c)	The file is updated.			
20.	Check the modified file into CVS by using the following command (all on one line): cvs commit -m "Added a second printf." JunoTest/test1.c	CVS accepts the new version of test1.c without error			
21.	Display the status the CVS repository using the following command: cvs status	CVS reports the new version (1.2) of test1.c			
22.	View a log of all CVS actions to date by issuing the following command: cvs log	CVS reports logged events.			
23.	Checkout the original version of test1.c and verify that the original copy is still in the repository by using the following commands: cvs checkout -r 1.1 JunoTest/test1.c more JunoTest/test1.c cvs release JunoTest (reply "y" to conformation of release)	The original version is checked out, the second printf command is shown not to be present, CVS prompts for confirmation of release			

Table 3.1 (cont.)

Step	Description	Expected Results	Comments	TC	QA
24.	Delete the JunoTest subdirectory and checkout the latest version of test1.c. Verify that the latest copy is still in the repository. Type: rm -rf JunoTest cvs checkout JunoTest/test1.c more JunoTest/test1.c	The JunoTest subdirectory is deleted, then is recreated with the updated version (the second printf statement is included in the file)			
25.	Checkout the file “Makefile” and execute the make utility. Type: cvs checkout JunoTest/Makefile cd JunoTest make	The Makefile is checked out and test1.c is compiled without error			
26.	Execute the program test1. Type: test1	“Hello world Hello world again” is displayed on the screen			
27.	Release the JunoTest files checked out. Type: cd .. cvs release JunoTest (reply “y” to conformation of release)	CVS prompts for confirmation of release.			
28.	Logoff of workstations.	Login session is closed, Login prompt is displayed.			

Table 3.1 (cont.)

End Time: ____________

Signature Page: Test Case 3.1 - System Integration Build and Configuration Management

<Not required>

Quality Assurance **Date**

Test Conductor **Date**

Comments:

3.2 System Start-Up and SDE Consolidated Data Flow (Ammended)

The purpose of this test case is to demonstrate the process for taking the system from a powered off and/or quiescent state to a state in which data is flowing (end user ready to work). These capabilities were developed in conjunction with the Juno Development Environment Foundation, the Development Environment System Software, the Consolidated SDS, the Consolidated Data and the Reliable Messaging threads.

3.2.1 Test Description

3.2.1.1 Detailed description

This test follows a typical ops scenario for bringing the system up. Hardware startup, initial self test and operating system start will be demonstrated using vendor supplied procedures. Custom software that requires manual startup will then be demonstrated. Verification of successful system start will be the display of data at the HCI workstation.

3.2.1.2 Resource Requirements

3.2.1.2.1 Test Personnel

Personnel required include at a minimum a Test Conductor. Since this test will not result in formal “buy-off” of any requirements, a QA representative is not required. Skills required by test conductor (or designee) include knowledge of vendor operating procedures, including both standard user and system administrator procedures, knowledge of custom software operating procedures and familiarity with data flowing through the system.

3.2.1.2.2 Hardware

The following hardware is required:

- SDE1 networking equipment
- The SDS Gateway
- One Consolidated Systems Gateway
- The following SDE1 workstations: sde1boot, sde1net, sde1hci1, sde1hci3, sde1hci7, sde1ddp1, PCIII Support PC, PC GOAL PC

3.2.1.2.3 Software

The following Software is required:

- Vendor supplied operating systems
- All custom software that is part of the Juno delivery

3.2.1.2.4 Data

Data required for this test includes input data streams from SDS, GMS and Metro.

3.2.1.3 Requirements Trace

This test will simulate the demonstration of compliance with System Level Specification requirements covering the following functions:

- Automated system start
- System diagnostics (power-on self test)
- Transfer of data from gateways to HCI workstations
- Basic OS commands/services:
  - OS boot
  - System shutdown

3.2.2 Pass/Fail Criteria

Successful completion of the test procedures without any problems that would result in the generation of critical PR’s and without an excessive number of major PR’s will be sufficient for this test to be considered passed.

3.2.3 Procedure

Refer to Procedure 3.2 in Table 3.2 for test procedures.

Procedure 3.2 - System Start-Up and SDE Consolidated Data Flow

Date: **Location:** **Start Time:**

Test Setup/Initial Conditions - This test will be executed in the SDE1 environment. All required software and hardware will be available in the same configuration they would be if the system were to have been powered down after support (e.g., for maintenance). Data sources (GMS, Metro, SDS) for the SDS gateway will be up and sending data, the SDS Gateway itself will also be up and sending data.

Step	Description	Expected Results	Comments	TC	QA
1.	Turn on SDE1 network hardware and PC's	Blinky lights start blinking on the network devices, PC's execute power on self tests, boots OS			
2.	Turn on the sde1net workstation, wait for it to finish booting (login screen will be displayed), then turn on the sde1boot workstation and wait for it to finish booting.	POST (Power On Self Test) tests occur, Operating system start procedures initiate, Login screens appear.			
3.	Turn on all remaining HCI workstations and the sde1ddp1 machine.	POST (Power On Self Test) tests occur, Operating system start procedures initiate, Login screens appear.			
4.	Turn the Consolidated Systems Gateway on.	Gateway executes power on self tests, boots OS, Gateway boots in approximately 45 seconds.			
5.	At the PCIII Support PC, enter: pcg98	Command is accepted, the user is prompted to select a Shuttle Data Stream (SDS)			
6.	At the PCIII Support PC, enter the number for the desired SDS. Type: 11	Command is accepted, data flow from the PCIII begins.			

Table 3.2

Step	Description	Expected Results	Comments	TC	QA
7.	At the PC GOAL PC, start the PC GOAL software to monitor the SDS data. At the D: prompt, type: wsm <CR> <CR>	The PC GOAL software starts and prompts user for input			
8.	At the PC GOAL PC, select the SDS data stream (sa083a FIFO), Type: T	Command is accepted, user is prompted for input			
9.	At the PC GOAL PC, select the display-build to be used. Type: I <CR> <CR> <CR>	Command is accepted			
10.	Login to the sde1hci7 workstation. When prompted, select the ground activity.	Login is accepted, a console window is opened, two xterm windows are opened, dnav software starts successfully.			
11.	At the sde1hci7 workstation, telnet into the CS gateway. Type: telnet sde1csg1	telnet connection is opened, the "->" prompt is displayed.			
12.	Initialize the gateway ATM interface. At the "->" prompt, type: <atm (wait for the message "fa0: switch=f21d0b9, port-2 assigned by switch") <CR>	Command is accepted, messages are displayed, "->" prompt is displayed.			

Table 3.2 (cont.)

Step	Description	Expected Results	Comments	TC	QA
13.	Start generating data via RTCN Type : <startmet (wait for the message "TCID Name Change to 'SA083A' for input Data Stream 20 Complete") <CR>	Command is accepted, messages are displayed, "->" prompt is displayed.			
14.	Log out of the gateway. At the "->" prompt, type: logout	Telnet session is ended.			
15.	Open the System messages window at the sde1hci7 workstation. In the Dnav master menu, select "System messages"	The System messages window opens and displays system messages			
16.	Initiate data acquisition at the sde1hci7 workstation. In the Dnav master menu, select "Global Apps", then select "Start receive process", then select "GW to HCI JUNO_DDP_8"	The System messages window indicates that the Start receive process is executing, no unexpected errors are displayed in the console window.			
17.	Start data display. In the Dnav master menu, select "Shuttle", then select any of the following: Wind Speed Wind Direction PAD A Wind Direction PAD B Temperature	The command is accepted (as shown in the System messages and console windows), the appropriate display(s) are started and are regularly updated.			
18.	Stop data display at the workstation. Select quit from display menu(s)	Display windows are closed.			
19.	Stop data acquisition at the workstation. In the Dnav master menu, select "Global Apps", then select "Stop receive process"	The command is accepted (as shown in the System messages and console windows), data acquisition is stopped			

Table 3.2 (cont.)

Step	Description	Expected Results	Comments	TC	QA
20.	logout of the sde1hci7 workstation.	Logout is successful, login prompt is displayed.			
21.	log in to each UNIX based workstation as root and issue the following command: shutdown -g0 -i0 shutdown the sde1net workstation last	Each workstation performs a clean shutdown and is ready to be powered off.			
22.	Power off each UNIX based workstation, the PCIII support PC and the PC GOAL PC	Power is off.			
23.	Power off the CS Gateway	Power is off.			
24.	Power off the network equipment	Power is off, the system is in the same configuration as in the beginning of the test.			
25.	Repeat steps 1 through 4	Same as previous iteration.			

Table 3.2 (cont.)

End Time: ____________

Signature Page: Test Case 3.2 - System Start-Up and SDE Consolidated Data Flow (Amended)

<Not required>

Quality Assurance **Date**

Test Conductor **Date**

Comments:

3.3 Test Case 3.3 - LCC-X Consolidated Data Flow

The purpose of this test case is to demonstrate the flow of consolidated data into the Consolidated SDS gateway, through the system to the end user display in the LCC-X environment. This test will also demonstrate system startup in the LCC-X. These capabilities were developed in conjunction with the Juno Consolidated SDS thread.

3.3.1 Test Description

3.3.1.1 Detailed description

Demonstrate the flow of consolidated data into the Consolidated gateway then through the system to the end users' display. This will be demonstrated in the LCC-X demonstration facility.

3.3.1.2 Resource Requirements

3.1.1.2.1 Test Personnel

Personnel required include at a minimum a Test Conductor. Since this test will not result in formal "buy-off" of any requirements, a QA representative is not required. Skills required by test conductor (or designee) include knowledge of vendor operating procedures, knowledge of custom software operating procedures and familiarity with data flowing through the system.

3.3.1.2.2 Hardware

The following hardware is required:

- The SDS Gateway (and associated Sun workstation)
- One Consolidated Systems Gateway
- LCC-X workstation

3.3.1.2.3 Software

The following Software is required:

- Vendor supplied operating systems
- All custom software that is part of the Juno delivery

3.3.1.2.4 Data

Data required for this test includes input data streams from SDS, GMS and Metro.

3.3.1.3 Requirements Trace

This test will simulate the demonstration of compliance with System Level Specification requirements covering the following functions:

- Automated system start
- System diagnostics (power-on self test)
- Transfer of data from gateways to HCI workstations
- Basic OS commands/services:
  - OS boot

3.3.2 Pass/Fail Criteria

Successful completion of the test procedures without any problems that would result in the generation of critical PR’s and without an excessive number of major PR’s will be sufficient for this test to be considered passed.

3.3.3 Procedure

Refer to Procedure 3.3 in Table 3.3 for test procedures.

Procedure 3.3 - LCC-X Consolidated Data Flow

Date: **Location:** **Start Time:**

Test Setup/Initial Conditions - This test will be executed in the LCC-X demonstration facility. All required software and hardware will be available in the same configuration they would be if the system were to have been powered down after support (e.g., for maintenance). Data sources (GMS, Metro, SDS) and the Sun workstation for the SDS gateway will be up.

Step	Description	Expected Results	Comments	TC	QA
1.	Turn on LCC-X network hardware and workstations	Blinky lights start blinking on the network devices, workstations execute power on self tests, boot OS, initial login prompt appears.			
2.	Turn the SDS Gateway and a CS gateway on and wait for the SDS GWY MAIN MENU to appear.	Gateway executes power on self tests, boots OS, SDS GWY MAIN MENU appears.			
3.	Decide which FEPC will be used. At the SDS Gateway, select the GWY STATUS from the MAIN MENU	GWY STATUS menu item is highlighted, next menu is displayed.			
4.	Go to RESOURCES HEALTH & STATUS and select YES	RESOURCES HEALTH & STATUS menu item is highlighted, next menu is displayed.			
5.	Go to any of the SBCs in slot 2 through 5 and select YES,	A screen similar Screen A is displayed, the status of all SBC's will be: Health Count: 00000000 Ready: No Tables Loaded: No			
6.	Login to the Sun workstation ("atlantis")	CDE menu appears.			
7.	Open an X terminal window.	Xterm starts			
8.	Telnet to the GCP using the new X terminal **(**NOTE: 2 gateways available: lccsdsg1 and lccsdsg2) telnet lccsdsg1	The "GCP0>" prompt is displayed			

Table 3.3

Step	Description	Expected Results	Comments	TC	QA
9.	Execute the SDS initialize string . Type (all on one line): sdsinit “A”, “2.34”, “1.15”, “1.16”, “1.26”, “A”, “gms1.dat”, “gms2.txt”, “11”, “20”	The following messages should be displayed: SDS init Initiated… Loading and initializing FEPCA…. Waiting for the FEPCA to be ready …. SDS initialized successfully…. A screen similar to Screen A will be displayed on the SDS gateway.			
10.	Logout from the GCP. Type: logout	Telnet session ends			
12.	At the Sun workstation, telnet into the CS gateway. Type: telnet lcccsg<x> where <x> is 1 or 2, depending on which gateway is used	telnet connection is opened, the “->” prompt is displayed.			
13.	Initialize the gateway ATM interface. At the “->” prompt, type: <atm (wait for a message similar to “fa0: switch=f21d0b9, port-12 assigned by switch”) <CR>	Command is accepted, messages are displayed, “->” prompt is displayed.			

Table 3.3 (cont.)

Step	Description	Expected Results	Comments	TC	QA
14.	Start generating data via RTCN Type : <startmet (wait for the message "TCID Name Change to 'SA083A' for input Data Stream 20 Complete") <CR>	Command is accepted, messages are displayed, "->" prompt is displayed.			
15.	Log out of the gateway. At the "->" prompt, type: logout	Telnet session is ended.			
16.	Logout of the Sun workstation	Login prompt appears.			
	Start HCI applications				
17.	Login to an HCI workstation as root	Login is successful, a console and an xterm window are started.			
18.	In an xterm window, change current userid and set up the workstation. Type: su - demo csh cd /usr/local/gms source gms_init	Commands are accepted without error.			

Table 3.3 (cont.)

Step	Description	Expected Results	Comments	TC	QA
19.	In the same window, remove the TCID table. If data for STS-83 is in use, Type: cd /usr/people/demo/hci.sts83/fd_hci_build/Exec fd_table_remove If data for STS-84 is in use, Type: cd /usr/people/demo/hci/fd_hci_build/Exec fd_table_remove	The command is accepted and executed without error			
20.	In the same xterm window, build the TCID table. If data for STS-83 is in use, Type: fd_table_build ../Static/sa083a.bsd If data for STS-84 is in use, Type: fd_table_build ../Static/sa084a.bsd	The table build completes without error.			
21.	In the same xterm window, start the receive process. If data for STS-83 is in use, Type: cd /usr/people/demo/hci.sts83/fd_hci_recv/Exec fd_hci_recv -rm JUNO_DDP_8 If data for STS-84 is in use, Type: cd /usr/people/demo/hci/fd_hci_recv/Exec fd_hci_recv -rm JUNO_DDP_8	The receive process is started without error.			

Table 3.3 (cont.)

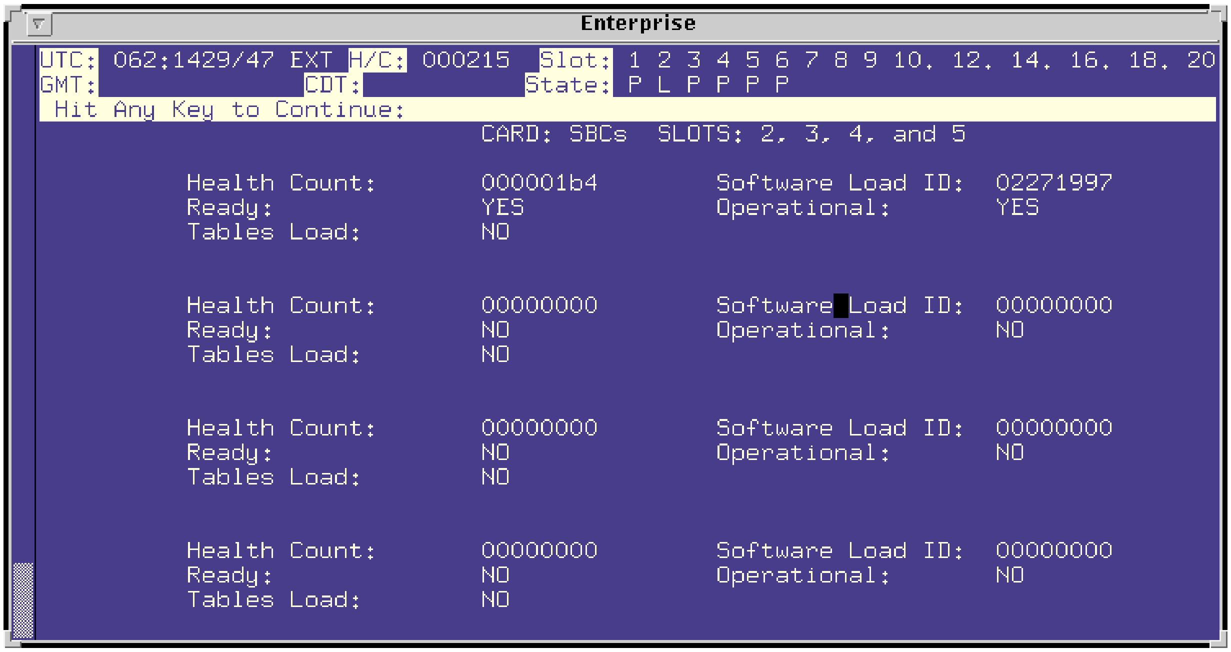
Step	Description	Expected Results	Comments	TC	QA
22.	In the same xterm window, start the display menu. If data for STS-83 is in use, Type: cd /usr/people/demo/hci.sts83/displays/Exec main-c If data for STS-84 is in use, Type: cd /usr/people/demo/hci/displays/Exec main-c	The display menu appears			
23.	Start the displays. From the display menu select: Monitor (then) Juno Demonstration Screens (then any of the following) ice team Relative Humidity Wind Speed Temp Wind Direction (followed by Pad A or Pad B)	Displays are started. Data is updated periodically.			
	Shutdown Data Displays				
24.	Stop data display at each HCI workstation: Select quit from display menu	Display is closed.			

Table 3.3 (cont.)

Step	Description	Expected Results	Comments	TC	QA
25.	Shutdown SDS Gateway: Login to the Sun workstation	CDE menu appears.			
26.	Start an X terminal from the CDE menu (if an xterm window is not already running)	xterm window opens			
27.	Telnet to the GCP using the new X terminal **(**NOTE: 2 gateways available: lccsdsg1 and lccsdsg2) telnet lccsdsg1	you should see the “GCP0>” prompt.			
28.	Execute the SDS terminate string: sdsterm “A”	The following messages should appear: SDS term Initiated… Notifying FEPCA to shutdown…. SDS terminated successfully….			
29.	Logout from the GCP logout				
30.	Logout of the Sun workstation	Login prompt appears.			

Table 3.3 (cont.)

End Time: ____________



```
UTC: 062:1429/47 EXT H/C: 000215  Slot: 1 2 3 4 5 6 7 8 9 10, 12, 14, 16, 18, 20
GMT:                 CDT:              State: P L P P P P
 Hit Any Key to Continue:
                              CARD: SBCs  SLOTS: 2, 3, 4, and 5

          Health Count:       000001b4       Software Load ID:  02271997
          Ready:              YES            Operational:       YES
          Tables Load:        NO


          Health Count:       00000000       Software Load ID:  00000000
          Ready:              NO             Operational:       NO
          Tables Load:        NO


          Health Count:       00000000       Software Load ID:  00000000
          Ready:              NO             Operational:       NO
          Tables Load:        NO


          Health Count:       00000000       Software Load ID:  00000000
          Ready:              NO             Operational:       NO
          Tables Load:        NO
```

Screen A

This screen shows the status of all four FEPCs. Slot 2, 3, 4, and 5. They are listed in that order. In this example the SBC in slot 2 (FEPCA) is in use.

Signature Page: Test Case 3.3 - LCC-X Consolidated Data Flow

<Not required>

______________________________________ _______

Quality Assurance **Date**

______________________________________ _______

Test Conductor **Date**

Comments:

____

____

____

____

____

____

____

____

____

3.4 Test Case 3.4 - System Operations and Basic Services

The purpose of this test case is to demonstrate a subset of basic system services required for operating the CLCS, specifically some of the COTS services required for operating a distributed UNIX based system that have not been tested in previous test cases. These capabilities were developed in conjunction with the Juno Development Environment Foundation thread. This is not intended to be a thorough checkout of all CLCS COTS software.

3.4.1 Test Description

3.4.1.1 Detailed description

Verify the following COTS components/functions: network tools, printing, file access control.

3.4.1.2 Resource Requirements

3.4.1.2.1 Test Personnel

Personnel required include at a minimum a Test Conductor. Since this test will not result in formal "buy-off" of any requirements, a QA representative is not required. Skills required by test conductor (or designee) include knowledge of vendor operating procedures, including both standard user and system administrator procedures, knowledge of COTS software.

3.4.1.2.2 Hardware

The following hardware is required:

- Two HCI workstations connected to the ethernet portion of the development environment
- Two workstations connected to the ATM portion of the development environment

3.4.1.2.3 Software

The following Software is required:

- Vendor supplied operating systems and associated utilities
- Three user accounts, two of which are members of the same group, the third not a member of that group

3.4.1.2.4 Data

A text file called test3.4 is required.

3.4.1.3 Requirements Trace

This test will simulate the demonstration of compliance with System Level Specification requirements covering the following functions:

- Network services
- Printing
- Security (file access controls)

3.4.2 Pass/Fail Criteria

Successful completion of the test procedures without any problems that would result in the generation of critical PR’s and without an excessive number of major PR’s will be sufficient for this test to be considered passed.

3.4.3 Procedure

Refer to Procedure 3.4 in Table 3.4 for test procedures.

Procedure 3.4 - System Operations and Basic Services

Date: **Location:** **Start Time:**

Test Setup/Initial Conditions - This test will be executed in the SDE1 environment. Workstations are available on each segment of the CLCS network and the account to be used is available on each of these workstations. The text file test3.4 is available in the home directory of the account to be used.

Step	Description	Expected Results	Comments	TC	QA
	Network services				
1.	Log onto a development workstation connected to the ethernet portion of the CLCS network	Login is successful, initial user display is presented.			
2.	telnet into another workstation connected to the ethernet portion of the network: telnet <workstation_id> <user_id> <password>	Remote login successful			
3.	Verify that the file test3.4 is not present at the remote workstation or if it is , delete it: ls -al /tmp rm /tmp/test3.4	The contents of the /tmp directory on the remote workstation is displayed			
4.	In a second xterm window, display the contents of the file test3.4 at the local workstation: cat test3.4	Contents of test3.4 are displayed			
5.	From the local workstation, transfer the file test3.4 to the other workstation using ftp. Type: ftp <workstation_id> cd /tmp put test3.4 quit	Transfer is successful			

Table 3.4

Step	Description	Expected Results	Comments	TC	QA
6.	Display the contents of the file test3.4 at the remote workstation. Type: cat test3.4	Contents of test3.4 are displayed			
7.	At the local workstation, verify that the file test3.4 is not present or if it is , delete it: ls -al /tmp rm /tmp/test3.4	The contents of the /tmp directory on the local workstation is displayed			
8.	At the remote workstation, transfer the file test3.4 to the local workstation using rcp. Type (all on one line): rcp /tmp/test3.4 <local_workstation_id>:/tmp/test3.4	Transfer is successful			
9.	Display the contents of the file test3.4 at the local workstation: cat /tmp/test3.4	Contents of test3.4 are displayed			
10.	Repeat steps 1-9 using two workstations attached to the ATM portion of the network. Use the ATM network.	The same capabilities demonstrated on the ethernet will be available using the ATM network.			
11.	Repeat steps 1-9 using one workstation attached to the ATM portion of the network and one workstation attached to the ethernet portion of the network. Use both network segments concurrently	The same capabilities demonstrated on the ethernet and ATM networks will be available across both (network traffic will be successfully routed from one segment to the other)			
12.	Execute a command on a remote workstation. Type: rsh <remote_workstation_id> ps -ef	A list of all processes currently executing at the remote workstation will be displayed.			

Table 3.4 (cont.)

Step	Description	Expected Results	Comments	TC	QA
	Printing services				
13.	From a workstation connected to the ethernet portion of the network, print the file test3.4: lpr test3.4	File is printed			
14.	Print the file test3.4 to a different printer: lpr -P<printer_name> test3.4	File is printed			
15.	Log off of all workstations	Session is closed, login prompts are displayed			
	File Access Controls				
16.	Log into a workstation with a user id that is included in the lor_st group. Do not use the testa user id.	Login is successful, initial user display is presented.			
17.	Create a subdirectory called test. Type: mkdir test	The subdirectory test is created.			
18.	Change directories to test, create a short script and check the permissions on the script and directory. Type: cd test vi test4 <insert> ps -ef <escape> :wq ls -al	The script test4 is created in the test directory. The owner of the test directory (“.” in the ls listing) is the username entered at login time and the group of the test directory is lor_st. The permissions on the test directory are set as: drwxr-xr-x The owner of the test4 script is the username entered at login time and the group of the test directory is lor_st. The permissions on the test4 script are set as: -rw-r--r--			

Table 3.4 (cont.)

Step	Description	Expected Results	Comments	TC	QA
19.	Attempt to execute the script test4. Type: test4	The script will not run, permission denied			
20.	Change the permissions on test4 and execute the script. Type: chmod 755 test4 ls -al test4	The permissions on test4 are changed to: -rwxr-xr-x The script executes successfully (the list of processes currently running will be displayed).			
21.	In another xterm window, change the username to testa. The testa user id is included in the lor_st group. Change directories to the test subdirectory. Type: su testa <password> whoami cd test pwd	The username is changed to testa and the current working directory is the test subdirectory of the home directory of the username from the original login session.			
22.	Execute the script test4. Type: test4	The script executes successfully (the list of processes currently running will be displayed).			
23.	In the first window, change the permissions on the test4 script and execute it. Type: chmod 754 test4 ls -al test4	The permissions on test4 are changed to: -rwxr-xr-- The script executes successfully (the list of processes currently running will be displayed).			

Table 3.4 (cont.)

Step	Description	Expected Results	Comments	TC	QA
24.	In the second window, execute the test4 script. Type: test4	The script executes successfully (the list of processes currently running will be displayed).			
25.	In the first window, change the permissions on the test4 script and execute it. Type: chmod 744 test4 ls -al test4	The permissions on test4 are changed to: -rwxr--r-- The script executes successfully (the list of processes currently running will be displayed).			
26.	In the second window, attempt to execute the test4 script. Type: test4	The script will not execute, permission denied.			
27.	In the first window, change the permissions on the test4 script and execute it. Type: chmod 754 test4 ls -al test4	The permissions on test4 are changed to: -rwxr-xr-- The script executes successfully (the list of processes currently running will be displayed).			
28.	In the second window, execute the test4 script. Type: test4	The script executes successfully (the list of processes currently running will be displayed).			

Table 3.4 (cont.)

Step	Description	Expected Results	Comments	TC	QA
29.	In the second window, change the username to testb. The testb user id is not included in the lor_st group. Change directories to the test subdirectory. Type: exit su testb <password> whoami cd test pwd	The username is changed to testb and the current working directory is the test subdirectory of the home directory of the username from the original login session.			
30.	In the second window, attempt to execute the test4 script. Type: test4	The script will not execute, permission denied.			
31.	In the first window, change the permissions on the test4 script and execute it. Type: chmod 755 test4 ls -al test4	The permissions on test4 are changed to: -rwxr-xr-x The script executes successfully (the list of processes currently running will be displayed).			
32.	In the second window, execute the test4 script. Type: test4	The script executes successfully (the list of processes currently running will be displayed).			

Table 3.4 (cont.)

Step	Description	Expected Results	Comments	TC	QA
33.	In the first window, change the permissions on the test4 script, change the group ownership to a group that the testb user id is a member of, and execute it. Type: chmod 754 test4 chgrp lor_cma test4 ls -al test4	The permissions on test4 are changed to: -rwxr-xr-- The group ownership of test4 is lor_cma The script executes successfully (the list of processes currently running will be displayed).			
34.	In the second window, execute the test4 script. Type: test4	The script executes successfully (the list of processes currently running will be displayed).			
35.	In the second window, change the username to testa. The testa user id is not included in the lor_cma group. Change directories to the test subdirectory. Type: exit su testa <password> whoami cd test pwd	The username is changed to testa and the current working directory is the test subdirectory of the home directory of the username from the original login session.			
36.	In the second window, attempt to execute the test4 script. Change the current directory to the next higher level directory. Type: test4 cd .. pwd	The script will not execute, permission denied. The current working directory is changed.			

Table 3.4 (cont.)

Step	Description	Expected Results	Comments	TC	QA
37.	In the first window, change the current directory to the next higher level directory. Change the permissions on the test directory. Type: cd .. chmod 754 test ls -al	The permissions on test are changed to: drwxr-xr--			
38.	In the second window, list the contents of the test directory. Type: ls -al test	The contents of the test directory are displayed.			
39.	In the second window, change the username to testb. The testb user id is not included in the lor_st group. Type: exit su testb <password> whoami pwd	The username is changed to testb and the current working directory is the home directory of the username from the original login session.			
40.	In the second window, attempt to list the contents of the test directory. Type: ls -al test	The command fails, permission denied.			
41.	In the first window, change the permissions on the test directory. Type: chmod 755 test ls -al	The permissions on test are changed to: drwxr-xr-x			

Table 3.4 (cont.)

Step	Description	Expected Results	Comments	TC	QA
42.	In the second window, list the contents of the test directory. Type: ls -al test	The contents of the test directory are displayed.			
43.	In the second window, attempt to write to the test subdirectory. Type: cp /etc/printcap test/printcap	The command fails, permission denied.			
44.	In the second window, attempt to delete the test4 script. Type: rm test/test4 (answer “y” if prompted to override mode 0754)	The command fails, permission denied.			
45.	In the first window, change the permissions on the test directory. Type: chmod 777 test ls -al	The permissions on test are changed to: drwxrwxrwx			
46.	In the second window, write to the test subdirectory. Type: cp /etc/printcap test/printcap ls -al test	The copy command is successful, the contents of the test directory are displayed.			

Table 3.4 (cont.)

Step	Description	Expected Results	Comments	TC	QA
47.	In the second window, end the testb user id session., delete the test directory and log out of the workstation. Type: exit rm -rf test <select logout from the pop-up menu>	The testb session is ended, the directory is deleted, the login session is ended and the login prompt is displayed.			

Table 3.4 (cont.)

End Time: ____________

Signature Page: Test Case 3.4 - System Operations and Basic Services

<Not required>

____ ________

Quality Assurance **Date**

____ ________

Test Conductor **Date**

Comments:

Appendix A - Acronyms and Definitions

AT	Acceptance Test - Test to accept hardware and software from a vendor
Certification	Final approval to use a system for a specified set of operations (e.g., hazardous operations in the HMF, launch operations, etc.)
CI	Configuration Item
CIT	CSCI Integration Test
CLCS	Checkout and Launch Control System
CM	Configuration Management
COTS	Commercial Off The Shelf
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
DAR	Delivery Acceptance Review
EDL	Engineering Development Laboratory
GSE	Ground Support Equipment
HCI	Human Computer Interface
HMF	Hypergol Maintenance Facility
HW	Hardware
HWCI	Hardware Configuration Item
IDE	Integrated Development Environment
I/F	Interface
KSC	Kennedy Space Center
LAN	Local Area Network
LCC	Launch Control Complex
LMSMS	Lockheed Martin Space Mission Systems and Services
LPS	Launch Processing System
NASA	National Aeronautics and Space Administration
MSC	Mission Systems Contract (held by LMSMS)
OS	Operating System
PTR	Post-Test Review
PR	Problem Report

QA	Quality Assurance
QE	Quality Engineering
QT	Qualification Test
RLV	Reusable Launch Vehicle
RTPS	Real Time Processing System
RVM	Requirements Verification Matrix
SDC	Shuttle Data Center
SDE	Satellite Development Environment
SEMP	System Engineering Management Plan
SFOC	Space Flight Operations Contract (held by USA)
ST	System Test
SLWT	Super Light Weight Tank
S&MA	Safety and Mission Assurance (includes Reliability, Maintainability, Safety and Quality Assurance)
STS	Space Transportation System
SW	Software
TC	Test Conductor
TPR	Test Progress Review
TR	Test Report
TRR	Test Readiness Review
UAT	User Acceptance Test - Test performed by user community post delivery as part of the system certification process
UIT	Unit Integration Test
USA	United Space Alliance
UT	Unit Test
Validation	Testing performed by organization(s) outside of the developing organization to ensure that the delivered system processes data correctly and conforms to the operations concepts

Appendix B - Requirements Traceability and Test Methods Matrix

The following table is intended to show which System Level Specification requirement is demonstrated in each CLCS System Test (at least one System Test per delivery) and what test method was used in that test case. This table will be updated and baselined with each System Test starting with the Redstone Delivery System Test.

Requirement	System Test	Test Case	Associated Threads	Test Method			
				Inspection	Analysis	Demonstration	Test

Appendix C - Resource Requirements

<TBS, defines personnel requirements, hardware and software configurations, data and test tools for standardized and generic test configurations. Not needed for Juno, may or may not be needed in future System Test procedures documents>

Appendix D - Standard Test Operating Procedures

<TBS, defines frequently used test or test setup procedures. Not needed for Juno, may or may not be needed in future System Test procedures documents>